Library Progress International *Print version ISSN 0970 1052*
Vol.44 No.3, Jul-Dec 2024: P.2431-2446 *Online version ISSN 2320 317X*

Original Article *Available online at www.bpasjournals.com*

Circuit Cipher Text - Policy Attribute - Based Encryption with Verifiable Delegation by using Cloud Computing

[1]V.Swapna, [2]R.Suhasini, [3]P.Sumathi,[4]V.Bhavani

[1]Assistant Professor, Department of Computer Science & Engineering, Keshav Memorial Engineering College, Hyderabad.
[2]Assistant Professor, Department of Computer Science & Engineering, CMR College of Engineering & Technology, Hyderabad.
[3]Assistant Professor, Department of Computer Science & Engineering(AI&ML), Institute of Aeronautical Engineering, Hyderabad.
[4]Assistant Professor, Department of Computer Science & Engineering(AI&ML), Keshav Memorial Engineering College, Hyderabad.

How to cite this article: V.Swapna, R.Suhasini, P.Sumathi, V.Bhavani (2024). Circuit Cipher Text - Policy Attribute - Based Encryption with Verifiable Delegation by using Cloud Computing. *Library Progress International,* 44(3), 2431-2446.

ABSTRACT
To keep data secure and under control when stored in the cloud, data owners might encrypt it using attribute-based encryption. Users with less powerful computers often utilize cloud servers to decrypt tasks in order to save computing expenses. Because of this, attribute-based encryption with delegation comes into being. Nevertheless, there are still certain restrictions and unresolved problems about the past relevant work. If you delegate encryption to a cloud server, there's a chance the service may change rewrite the supplied content, provide a false calculation result. In an effort to save costs, they may mistakenly treat qualified customers as ineligible. In addition, there's a chance that the access limits won't be flexible enough while encryption is underway. Given that the most stringent form of access control may be achieved with a policy for generic circuits, our research takes into account a framework for constructing hybrid encryption using circuit cipher and text-policy attributes, with verifiable delegation.Not only do such a system and verifiable computation ensure that delegated computing outputs are accurate, but they also guarantee protected information and fine-grained access control. Our method also protects against chosen-plaintext assaults, as long as you use the Deterministic k-multilinear style We assume Diffie-Hellman algorithm. The feasibility and efficacy of the proposed solution are further confirmed by an exhaustive simulation exercise.

Keywords: Malicious, Encryption, Cipher text, Confindentiality.

INTRODUCTION

The outcome is the emergence of attribute-based encryption with delegation. However, previous efforts in this area still have several unanswered questions and limitations. As an example, the delegated encryption text might be changed or replaced by the cloud servers during delegation, leading to a purposely generated computational result. In an effort to save money, they can react to qualified consumers as ineligible, which might lead them astray. On top of that, the access limits could not be flexible enough to accommodate changes made during encryption. In this study, we take a look at a methodology for developing hybrid encryption with text-policy attributes that makes use of circuit ciphers and verifiable delegation. This is due to the fact that the most stringent kind of access control may be enabled by expanding circuit regulations.

1.2. OBJECTIVE

On the other hand, we execute our scheme across the whole numbers. The scheme's feasibility in cloud computing is shown by the computation and transmission expenses. Therefore, we may utilize it to provide verified

delegation, granular access control, and cloud data confidentiality.

1.3 Existing System:

The allocated ciphertext may be changed or replaced on modern cloud servers, and they can return with a maliciously faked compute result. They may possibly deceive eligible customers by responding to them as ineligible as a cost-saving approach. Furthermore, it's possible that the access restrictions won't be adequately adaptable throughout the encryption process.

1.4 Proposed System

It is assumed that the suggested system is secure in line with what is supposedly the Diffie-Hellman k-multilinear decision-making theory. Conversely, we carry out our plan on an integer scale. The computation and transmission expenditures demonstrate the scheme's viability in cloud computing. This means it might be useful for ensuring the privacy of cloud data, granular access control, and validated delegation. Since generic circuit policies allow for the most stringent kind of access control to be implemented, our study investigates a way to generate encryption using a hybrid circuit and ciphertext policy with attributes with verifiable delegation. Data confidentiality, granular access control, and accurate delegated computing outputs are all guaranteed by the system using the encrypt-then-mac technique and verified computation. Furthermore, our method prevents select-plaintext attacks that use the k-multilinear Decisional Diffie-Hellman theory.

.

1.4.1 Proposed System Advantages

- One possible way to recognize trustworthy and safe outsourced stockpiling is by multi-istributed storage, which distributes information with some kind of repetition among several mists.

LITERATURE SURVEY

Outsourcing the decryption of ABE Ciphertexts

Author :M. Green, S. Hohenberger, B. Waters

Description :

The groundbreaking authentication using attributes (ABE) users are able to encrypt and decrypt data using the public key encryption technique. decode communications depending on user attributes. To illustrate the point, imagine if you wanted to encrypt data such that only people who fulfilled the conditions of ("Faculty" OR ("PhD Student" AND "Quals Completed")) could read it. A lot of storage and cloud computing applications are currently considering ABE due to its expressiveness. One major drawback about Could it be that the volume of ciphertext and the time needed to decipher it grow in proportion to the formula for access's complexity.

In this paper, we provide an ABE paradigm that greatly reduces this difficulty for customers. Let us pretend that ABE ciphertexts are kept in the cloud. To prevent the cloud from decoding any portion of the user's communications, we demonstrate how, using a single transformation key that the user may provide, the cloud can turn an ABE ciphertext that is filled with an individual's components into a ciphertext in the El Gamal manner with a fixed size.

Extensive performance measurements, an implementation of our algorithms, new constructs, and new security definitions for replayable CCA and CPA with outsourcing are all part of our effort to describe and prove the benefits of this method. Users may often reduce the volume of communications while also saving a lot of bandwidth and decryption time in a typical arrangement.

Attribute-based encryption with verifiable outsourced decryption

Author :J. Lai, R. H. Deng, C. Guan, J. Weng

Description :

Users may encrypt and decrypt data based on their own attributes using attribute-based encryption (ABE), a public-key based one-to-many encryption system. Through the use of access rules and assigned characteristics associated with private keys and ciphertexts, ABE has the ability to provide flexible access control for encrypted data stored in the cloud. The present state of ABE algorithms is inefficient in part because decryption requires costly pairing operations, the quantity of which increases with the access policy's complexity. A new ABE method was suggested by Green et al., which eliminates the cost worry for customers by using outsourced decryption.

In this setup, a user's attributes or access policy may be used by an untrusted server, such a cloud provider, to decipher an ABE ciphertext with a transformation key that the user provides. There is little processing complexity associated with decrypting ciphertext into plaintext. The inherent security protocols of an ABE system render any attempt to decrypt encrypted data futile, even if the target is a malicious cloud.

However, it does not imply that the cloud's transformation is accurate. Here, we look at verifiability, a new ABE criteria that makes use of outsourced decryption. The capacity to easily determine if the change was executed appropriately is what we mean when we talk about verifiability. In addition to the formal ABE paradigm with externally certified decryption, we also provide a specific technique. Without depending on random oracles, we demonstrate that our suggested approach is safe and verifiable. We conclude with the system's implementation and performance evaluation findings, which show that users may significantly reduce the computational resources they need to consume.

Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption

Author : Ming Li; Shucheng Yu; Yao Zheng; Kui Ren; Wenjing Lou

Description :

One new patient-centered paradigm for exchanging medical records that are often contracted to third parties, including cloud providers, is the personal health record (PHR). There have been several privacy issues, however, due to the fact that such third-party systems might potentially allow unauthorized persons to access critical health information. Patients should have control over who has access to their personal health records (PHRs) by encrypting them before outsourcing. Key management scalability, privacy exposure risks, flexible access, and quick user revocation are still the fundamental obstacles to attaining data access control with granularity and cryptography for security. To protect protected health information (PHI) kept on semitrusted servers, we provide a a set of data access limitations and a new architecture focused on patients.

With attribute-based encryption (ABE) methods, we can keep all patient health records safe and provide them granular control over who may access their data. For both owners and users, we streamline key management. alike by dividing PHR system users into separate security zones, which is distinct from previous research on the topic of secure data outsourcing. This approach works in situations when several data owners are involved. When many authorities deploy multiauthority ABE at the same time, patients' privacy is greatly protected.

In addition, our method allows for break-glass access in case of emergencies, dynamic alteration opportunity to rules or file characteristics, and rapid on-demand revocation of users or attributes. The efficiency, scalability, and security of our suggested approach are shown by extensive experimental and analytical findings.

A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing

Author : Zhiguo Wan; Jun'e Liu; Robert H. Deng

Description :

The concept of cloud computing has caused quite a stir in the information technology sector in recent years. As more and more businesses rely on cloud services to store and process sensitive data, worries about the privacy and security of outsourced data have grown. For controlling who can access data stored in the cloud, there are several attribute-based encryption (ABE) solutions available; however, the majority of these systems struggle to construct sophisticated access control lists (ACLs). As a flexible and scalable solution for managing access to data stored in the cloud, we provide hierarchical attribute-set-based encryption (HASBE). Hierarchical attribute-set-based encryption (HASBE) is an improvement on ciphertext-policy attribute-set-based encryption (ASBE).

With its hierarchical structure, the proposed method achieves scalability while retaining the compound properties of ASBE, such as fine-grained access control and flexibility. Furthermore, HASBE handles user revocation more deftly than current systems by using multiple value assignments for access expiry time.

We clearly guarantee HASBE's security by evaluating Bethencourt's CP-ABE method, which stands for ciphertext policy attribute-based encryption, which we call CP-ABE, for its efficiency and computational cost. Through rigorous testing, we demonstrate our system's flexibility and efficacy in handling access control for data that is outsourced on the cloud.

Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems

Author : Junbeom Hur; Dong Kun Noh

Description :

The two most challenging parts of data outsourcing are supporting rule updates and enforcing authorization conditions. Data owners may implement access control rules on outsourced data using encryption protocols based on ciphertext policies, which might be a cryptographic solution to these issues. When using outsourced architecture using attribute-based encryption, there are several problems with user and attribute revocation.

This article presents our findings— provide an attribute-based access control scheme based on ciphertext policies method that is both effective and user-revocation-based, allowing for the implementation of access control rules. a two-pronged approach to encryption that combines attribute-based encryption with selective group key distribution for granular access control. We demonstrate the suggested method for safely managing the outsourced data. The investigation's findings prove that the data outsourcing systems' proposed technique is both safe and effective.

3.1 SYSTEM ANALYSIS:

- **3.1.1..Existing System:**

The present cloud servers have the ability to alter or swap out the assigned ciphertext and return with a counterfeit compute result that is intended to do harm. For cost-saving measures, they could potentially react to eligible consumers as ineligible, misleading them. Moreover, the chance to controls may not be sufficiently flexible throughout the encryption process.

- **3.1.2 Existing System Disadvantages:**

Conversely, mystery sharing generates offers using random information seeds. In the unlikely event that customers implant unique, irregular data seeds.

They have distinct information, yet their shares will differ and cannot be reproduced.

- **3.1.3 Proposed System**

The security of the suggested system is shown by the k-multilinear Decisional Diffie-Hellman assumption. Conversely, we carry out our plan over the integers. The compute and transmission costs demonstrate the scheme's viability in cloud computing. As a result, we may use it to provide cloud data secrecy, granular access control, and verified delegation. Our study considers a construction for establishing circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation, because policy for generic circuits allows to accomplish the strongest type of access control. Verifiable computation, an encrypt-then-mac technique, and such a system provide not only the accuracy of the delegated computing outputs but also the secrecy of the data and fine-grained access control. Furthermore, our plan works under the k-multilinear Decisional Diffie-Hellman assumption to secure against chosen-plaintext attacks.

- **3.1.4 Proposed System Advantages**

Multi-istributed storage provides a feasible way to recognize trustworthy and safe outsourced stockpiling by dispersing information with some kind of recurrence throughout many mists.

3.3 FUNCTIONAL REQUIREMENTS

A software system's or one of its components' functions are defined by a functional requirement. A collection of inputs, behaviors, and outputs are used to characterize a function. The suggested system starts with a utility verification method that protects privacy and is built on cryptographic techniques for DiffPart, a differentially private scheme made for set-valued data.

3.4 NON-FUNCTIONAL REQUIREMENTS

EFFICIENCY

In this study, we suggest a novel structure termed. Verifiable computation, an encrypt-then-mac technique, and such a system provide not only the accuracy of the delegated computing outputs but also the secrecy of information and limited access. In addition, our strategy secures against attacks using chosen-plaintext the k-multilinear Decisional Diffie-Hellman assumption.

ObjectOriented

Any language may be considered object-oriented if it satisfies at least four requirements.

1. Inheritance: This is the technique of developing new classes based on the behavior of pre-existing ones by extending them in order to reuse pre-existing code and add new features as required.
2. Encapsulation: This is the process of fusing data and delivering an abstraction.

3. Polymorphism: As the name implies, polymorphism is the process of delivering distinct functionality via functions with the same name, depending on the method signatures.
4. Dynamic binding is a useful technique for developing code when we don't know an object's exact type. It is a method of giving a program the most functionality possible for that particular type at runtime.

- **3.6.1 Evolution of Collection Framework:**

  Nearly all Java collections are built on top of the java.util.Collection interface. The essential elements of every collection are outlined in collection. The interface provides the add() and remove() methods for adding to and removing from a collection, respectively. It is also essential to use the toArray() method, which converts the collection into a simple array containing every item in the collection. Last but not least, the includes() method ascertains if a certain element is included in the collection. Since the Collection interface is a subinterface of java.util.Iterable, the iterator() method is also accessible. There is an iterator for each collection that goes through each element in the collection. Additionally general is collection. Any collection may be written to store any class. For example, a collection may include strings inside of it, and its members can be utilized as strings without casting.

Three primary categories of collections exist:

- Lists: may be used in the same manner as regular arrays, are always arranged, and may include duplicates.
- Sets: their elements cannot be accessed randomly and cannot contain duplicates.
- Maps: link distinct keys and values, allow for arbitrary key access, and include the ability to store same values twice.

LIST

In order to create lists, the JCF makes use of the java.util.List interface. Simply said, a list is an array with more freedom. The parts are supposed to go in a certain order, and it's okay to have duplicates. You may arrange the elements anyway you want. Finding them on the list might be another option. Lists are implemented by two distinct classes. One of them, java.util, uses the list as an array.Here is the ArrayList. By rearranging the array's items, the class is able to perform list-specific actions. The java.util.LinkedList implementation is another choice. Using this class, the items are organized into nodes, and each node in the list has a reference to the node before and after it. Following the directions, one may add or delete things from the list by rearranging the points to put the node properly.

SET:

A package called java.util is available in Java.Sets are defined by their interfaces. Duplicate items cannot be included in a set. The collection is also not organized in any particular way. Because of this, finding items using an index is not possible. The Sets of hashes, linked hashes, and trees implemented on the Java platform classes all provide implementations of sets. Using a hash table is what HashSet does. It takes a java.util.HashMap and uses it to eliminate duplicates while preserving hashes and entries. The LinkedHashSet class in Java builds upon this idea by generating a doubly linked list that associates elements according to their insertion order. Doing so guarantees a consistent iteration sequence for the set. The red-black tree that is supplied by java.util.TreeMap is used by Java.util.TreeSet. To check for duplication, there is a red-black tree. As a result, Tree Set may also use the java.util implementation.A SortedSet that is a part of the Java.util package.The SortedSet interface extends This is a Java.util programme.It is used. The elements of a sorted set are sorted in a different way than those of a regular set. This sorting may be done either by including a method in the sorted set's constructor or by using the compareTo() function on each member. Starting and concluding with the initial or the end of the sorted set, as well as utilizing minimum and maximum values, are all ways to build subsets. The starting and ending points of the ordered set may be obtained. The java.util package implements the SortedSet interface.Array of Trees

You are looking at a Java.util application.Not only that, but the java.util.You may get another one via the NavigableSet interface. Equipped with a number of additional techniques compared to SortedSet. We traverse the set in search of an element close to the parameter using the top, bottom, floor, and ceiling methods. In addition,

the set comes with an iterator that can traverse the whole list of items. very much like SortedSet in Java.util.The NavigableSet is implemented using TreeSet.

MAP:

Java maps are defined via the java.util.Map interface. Values are linked to keys in maps, which are basic data structures. Elements have worth. This makes the map rather adaptable. Using the element's hash code as the key reduces the map to a set. It becomes a list when the numbers are strictly increasing. A total of three libraries: Java.util.HashMap, Java.util.LinkedHashMap, and Java.util.The TreeMap modules are responsible for implementing maps. One uses a hash table in HashMap. To find the data in different buckets, key hashes are used. A double linked list is constructed between the components in LinkedHashMap, which is an extension of this. Following the original sequence of their introduction to the map will allow you to access the parts. Using a black-and-red tree, TreeMap differs from HashMap and LinkedHashMap. Every node in the tree represents a value in the map, and every key represents a value at that node.

Thread:

Simply said, a thread is the execution route of a program. When several tasks or events must occur simultaneously, the single-threaded execution of most modern programs could lead to unexpected consequences. Think about what would happen if a program couldn't do things like read keystrokes or create visuals. Because it can't process more than one event at a time, the application can only focus on keyboard input. Two or more program parts running in tandem is the ideal way to fix this issue.

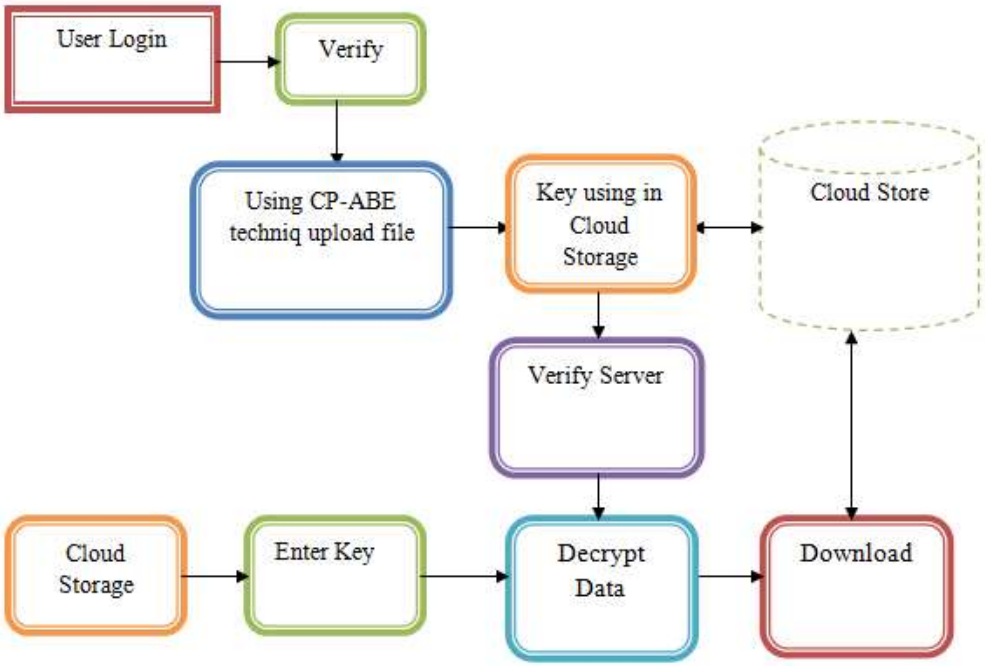
1.1 **Creating threads**

Two ways to introduce threads to Java have been helpfully offered by the designers: creating an interface and adding an extension to a class. Java extends a parent class to inherit its variables and functions. Considered here, one may only extend or inherit from a single parent class. Creating threads in Java is most often achieved via implementing interfaces, which circumvents this limitation. (Note that inheriting just permits the class to function as a thread. When to start() execution, etc., is determined by the class.)

Programmers may use interfaces to build a class's framework. They are used in the creation of the specifications required for the implementation of a certain set of classes. The interface only puts things up; all work is done by the class or classes that implement it. The same rules must be followed by all of the many classes that implement the interface.

Conclusion: Swing's inherent ability to override the GUI controls that the host operating system (OS) provides for it demonstrates the enormous degree of freedom that Swing has. Swatch "paints" using the 2D APIs in Java for its controls instead of requiring access to a native UI toolkit. Java's thread scheduler is rather simple. Each thread's priority may be dynamically changed by making executions of the setPriority() function. using the previously stated concepts in our project to guarantee the server runs well.

4.1 SYSTEM ARCHITECTURE:



4.2 DATA FLOW DIAGRAM:

Level 0:

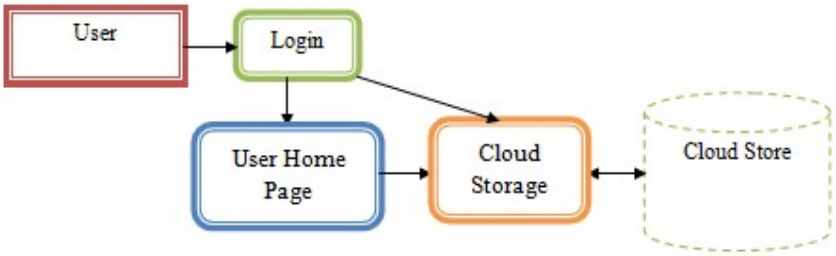


Figure: 4.2.1

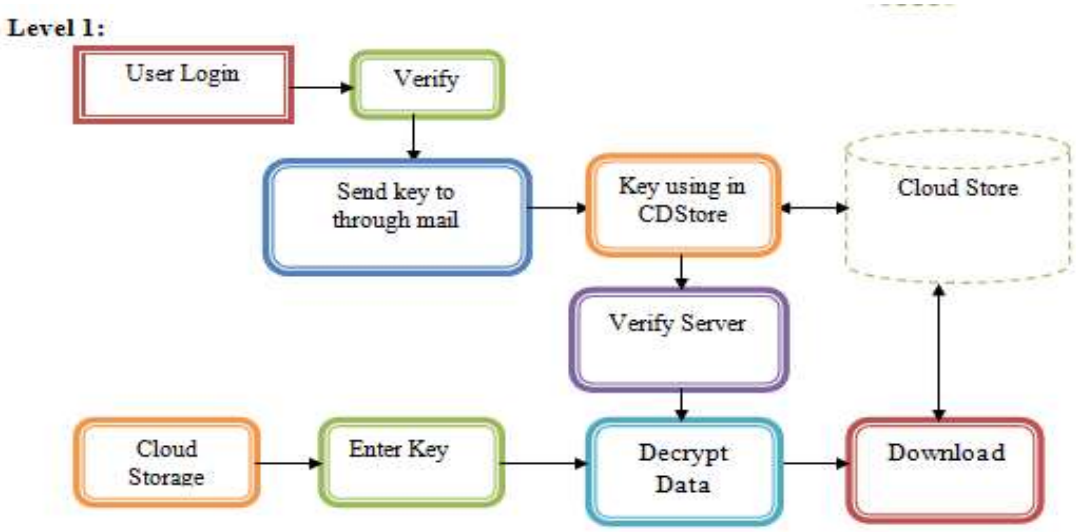


Figure:4.2.1

4.3.10 E-R DIAGRAM

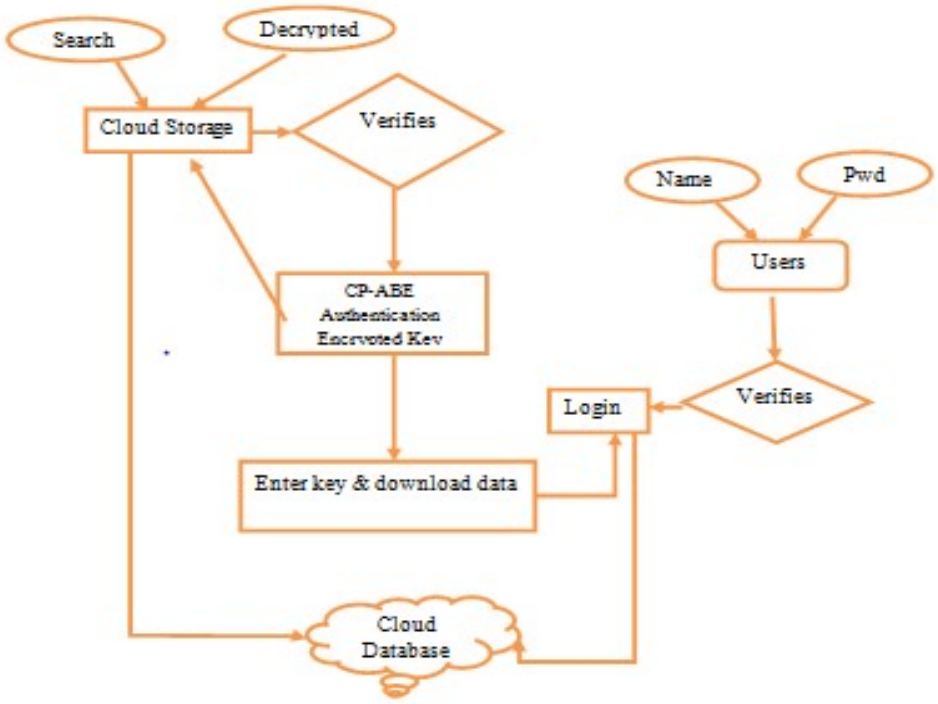


Figure:4.3.10

EXPLANATION:

The Entity-Relationship Model (ERM) provides a conceptual and abstract representation of data. A relational database is often the product of an approach to database modeling known as entity-relationship modeling. This method involves the creation of a conceptual schema or semantic data model of the system.

TECHNIQUE USED OR ALGORITHM USED

Ciphertext-policy attribute-based encryption

Both the idea of security and our hybrid VD-CPABE are defined by us. Verifiable delegation and fine-

grained access control are made possible by this system's use of circuit ciphertext-policy attribute-based encryption, an encrypt-then-mac method, and symmetric encryption. Combining the two algorithms, a hybrid VD-CPABE scheme is built using Establish, Hybrid-Encrypt, Key-Gen, Transform, and Verify-Decrypt. Every algorithm is described in depth below. • Placing (λ, n, l). You need the maximum depth (l), number of features (n), and security parameter (λ) of a circuit as inputs for this technique to be effective. The person in charge carries out this process. It produces the private master key (MK) as well as the public parameters (PK). further details. This paper has not been thoroughly revised, but it has been accepted for publication in a later edition of this journal. Content may change before it is released in its final version. A study on hybrid encryption in cloud computing with attribute-based and verifiable delegation utilizing circuit ciphertext-policy f, PK, M: Hybrid Encryption for 5 is released in the 2015 edition of the IEEE Transactions on Parallel and Distributed Systems with the DOI 10.1109/TPDS.2015.2392752. The data owner is in charge of this algorithm. Its two easily separated parts are the authenticated symmetric encryption (AE) and the key encapsulation method (KEM). – The public parameters PK and the circuit's access structure f are sent into the KEM algorithm. It selects a random string R first, then computes the complement circuit f. Next, we produce the CP-ABE ciphertext (CKM,CKR), where KM = {dkm, vkm} and KR = {dkr, vkr}. – A symmetric key, a random string R, and a message M are sent to the AE algorithm.

5.1 MODULES

This project having the following five modules:

- **User Interface**
- **Cloud Storage**
- **Security Model**
- **Cipher text - policy attribute-based encryption**
- **Hybrid encryption**
- **Email Authentication**

- **5.1.1 MODULE EXPLANATION**
  - **User Interface:**

Only by entering their username and password may users establish a connection with the server. If the user has already logged out, they may log in directly to the server; if not, they need to register using their email address, password, and username. The server will create an account for each user in order to keep up with the upload and download speeds. Name will be the user ID that is configured. A page may usually be accessed after signing in.

- **Cloud Storage**

Data kept in the cloud is organized into logical pools, with hosting companies often owning and managing the physical infrastructure, and storage space distributed across several servers in different locations. The upkeep of the physical infrastructure and the data's accessibility and availability are both the responsibility of these cloud storage providers. Data pertaining to end users, companies, or applications may be purchased or leased from suppliers of storage space.

- **Security Model**

We begin by discussing the security definition independently, as our hybrid VD-CPABE approach is built using both authenticated encryption (AE) and key encapsulation mechanism (KEM). Playing these games against adversary A helps to demonstrate the secrecy quality (IND-CPA, or the indistinguishability of encryptions against selected chosen plaintext assaults) that is necessary for KEM. Game.The Kem • That. The challenge access structure f∗ is provided by the opponent. * Initialization. After the opponent is given the PK, the Setup procedure is executed by the simulator. • KeGen The first inquiry. Attribute sets x1,..., xq1 are repeatedly queried by the attacker in search of secret keys. Make sure that f∏(xi) = 0 for all i in q1.

Make advantage of encryption. Following a random coin toss (b) and a random selection of K1 from key space, the simulator encrypts K0 using the structure f∏. Afterwards, the opponent receives Kb and the deciphertext CK∏ from the simulator. Asking for KeyGen II. Every time f■(x) = 0, the attacker keeps

asking for secret code for an different set of qualities (xq1,..., xq). • Think. The counter-party provides the benefit of a rival, b′, in place of b. is detailed below. The solution to this particular game is 1, 2. If there is no discernible benefit, a KEM method will be safe against well-planned plaintext assaults. The following games versus adversary A show how AE relies on the secrecy property, which is the ability of encryptions to be undetected against certain ciphertext assaults (IND-CCA). Have fun.AE • That. The enemy sends two messages, M0 and M1, of the same size. * Initialization. Simulator does Setup and creates symmetric key KAE. Implement a cryptographic system. A random coin, b, is flipped by the simulator. After that, it safely encrypts megabytes using the symmetric key KAE, creating the ciphertext C*h*, and then it delivers it to the enemy. • Decipher inquiries. The enemy keeps asking the same decoding questions over and over again. If the provided The encrypted "data C̃ = C∏" the adversary will get DKAE(C) and ϖKAE(C) from the simulator.

- **Ciphertext-policy attribute-based encryption**

We detail our hybrid VD-CPABE and the security mechanism it employs here. By using symmetric encryption, an encrypt-then-mac technique, and feature-based encryption using circuit ciphertext policies, the system ensures confidentiality, fine-grained access control, and verifiable delegation.

First, configure; second, generate keys; and last, change and Verify-Decrypt are the two algorithms that constitute system that combines VD and CPABE. Each algorithm is described below. • Placing (λ, n, l). In this method, which is carried out by the authority, the maximum depth (l), amount of features (n), and security parameter (λ) of a circuit are inputs. It creates the private master key (MK) in addition to the public parameters (PK). further details. Despite lacking a comprehensive edit, this material has been approved for publishing in an upcoming issue of the journal. The material may be revised before it is published in its final version. In the article "Hybrid Encryption for 5," published published in the journal IEEE Transactions on Distributed and Parallel Systems, the authors discuss the use of attribute-based and verifiable delegation in cloud computing, as well as circuit ciphertext policies. The paper has a DOI of 10.1109/TPDS.2015.2392752. The owner of the data controls this algorithm. Authenticated symmetric encryption (AE) and key encapsulation mechanism (KEM) are its two easily removable parts. - A The circuit's access structure f and the public parameters PK are inputs to the KEM algorithm. After selecting a random string R, it calculates the complement circuit f. The generation of the CP-ABE ciphertext (CKM,CKR) follows, where KM is defined as {dkm, vkm} and KR is defined as {dkr, vkr}. - A A message M, a symmetric key, and a randomly generated string R are the three inputs to the AE method.

- **Hybrid encryption**

For encryption, we use hybrid technology. Messages of any length may be encrypted using the ubiquitous KEM/DEM structure for hybrid encryption, which was established by Cramer and Shoup. One product of their innovative work is the KEM/DEM hybrid encryption idea, which merges symmetric encryption with a one-time MAC. Even better, this version can meet even more stringent security requirements. ABE Verification of Assignment. Since ABE originally appeared, several advances have been achieved. The utilization of compute outsourcing is a crucial approach. Researchers developed the first ABE using an external decryption process to reduce the computation cost during decryption. After then, Lai et al. presented the idea of ABE with verifiable outsourced decryption. Their goal in using a commitment is to guarantee the original ciphertext is legitimate. However, because the data owner doesn't provide any personally identifiable information while committing, the untrusted server may create a commitment for any message he wants. The ciphertext of the message may be changed in this way. Additionally, it is not enough to adjust the ciphertext's guarantees with respect to the message alone. Even if the user has the proper authorization, the cloud server might pose as Terminator TM to make them believe they don't have access to the data.

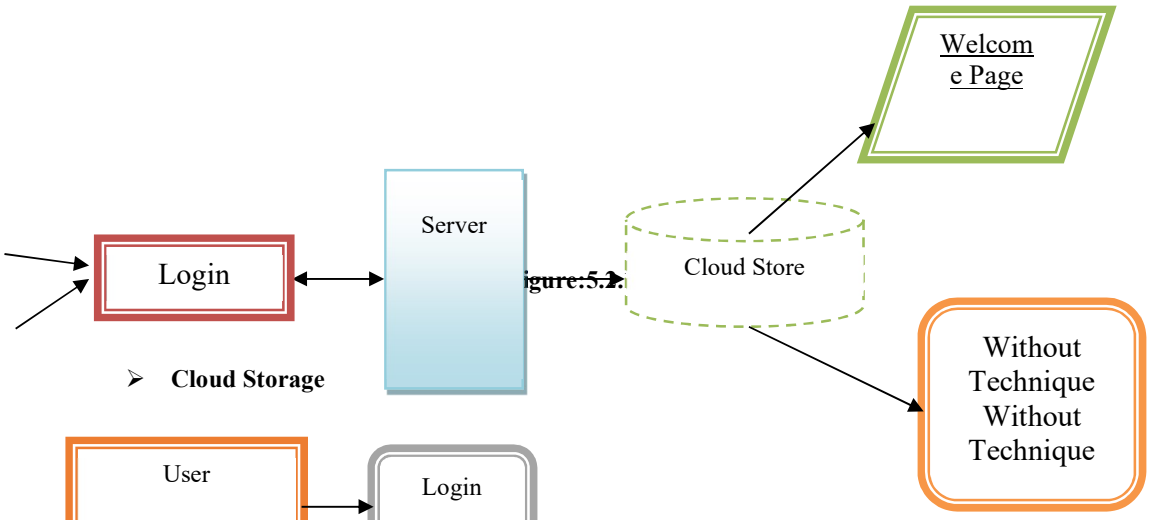
- **Email Authentication:**

  Providing messages delivered over the email transport system with information that can be independently verified is the aim of email authentication. It is a coarse-grained authentication that often occurs at the Administrative Management Domain (ADMD) level and denotes no form of authorization. Put differently, email authentication serves the purpose of confirming the identity of people engaged in a message transfer since the parties involved have the power to alter it. The results

of this validation may subsequently be applied to distribution options, which are entirely distinct and beyond the scope of email authentication.

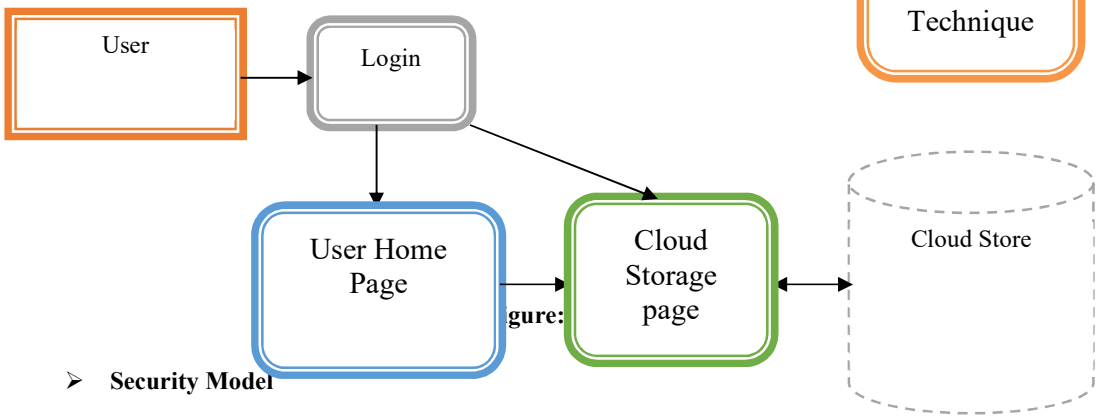
5.2 MODULE DIAGRAM:

- **User Interface**



igure:5.2.

- **Cloud Storage**



gure:

- **Security Model**

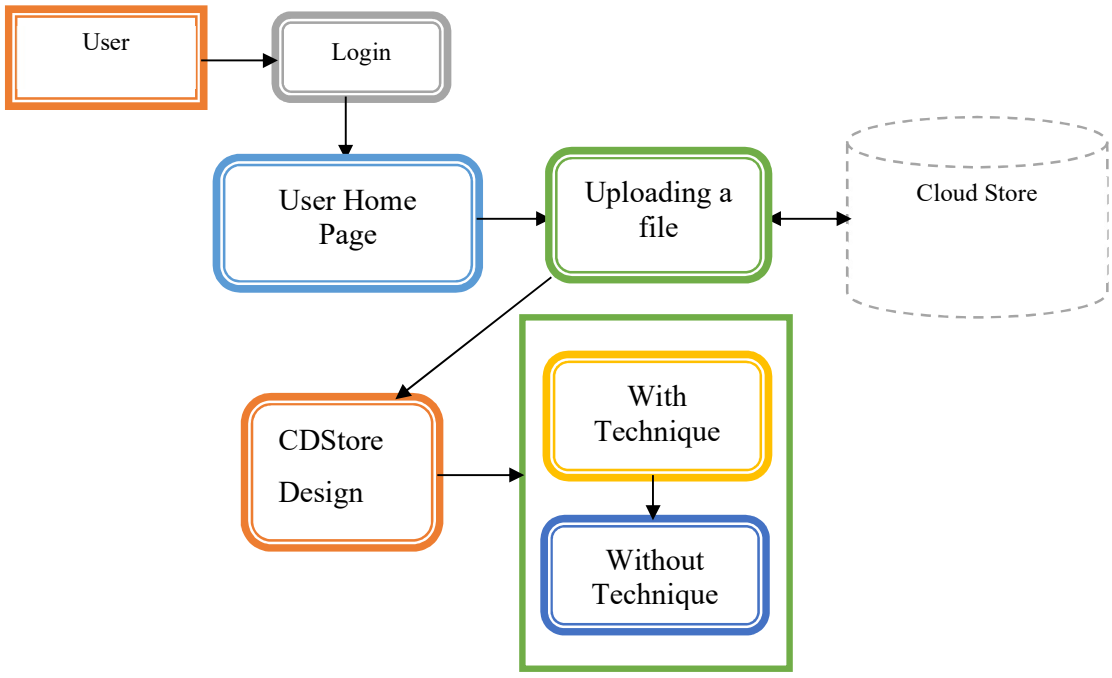


Figure: 5. 2.3

- **Cipher text - policy attribute-based encryption**

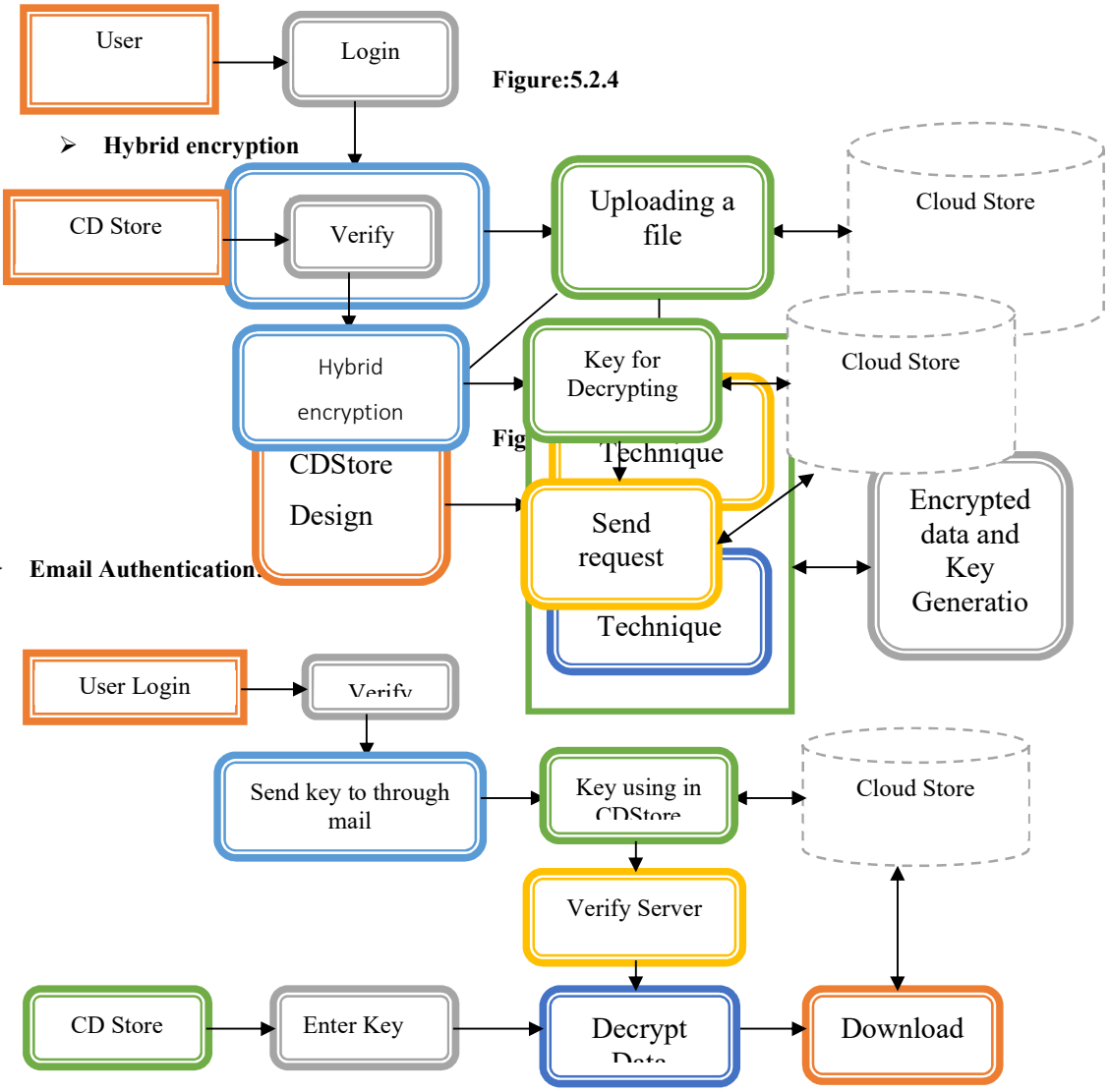


Figure:5.2.4

- **Hybrid encryption**

- **Email Authentication**

Figure:5.2.6

5.3 GIVEN INPUT EXPECTED OUTPUT:

- **User Interface**

Input: Put your password and login name here.
Output: Open the home page straight if the user is legitimate; if not, display an error message and refer the user to the registration page.

Cloud Storage

Input: Users that have enrolled with the same cloud storage service will be able to see other users.
Output: We will provide the customer the ability to see files uploaded by other cloud clients by allowing them to search the file names. Nonetheless, the user will be able to see the data in an encrypted format.

- **Security Model**

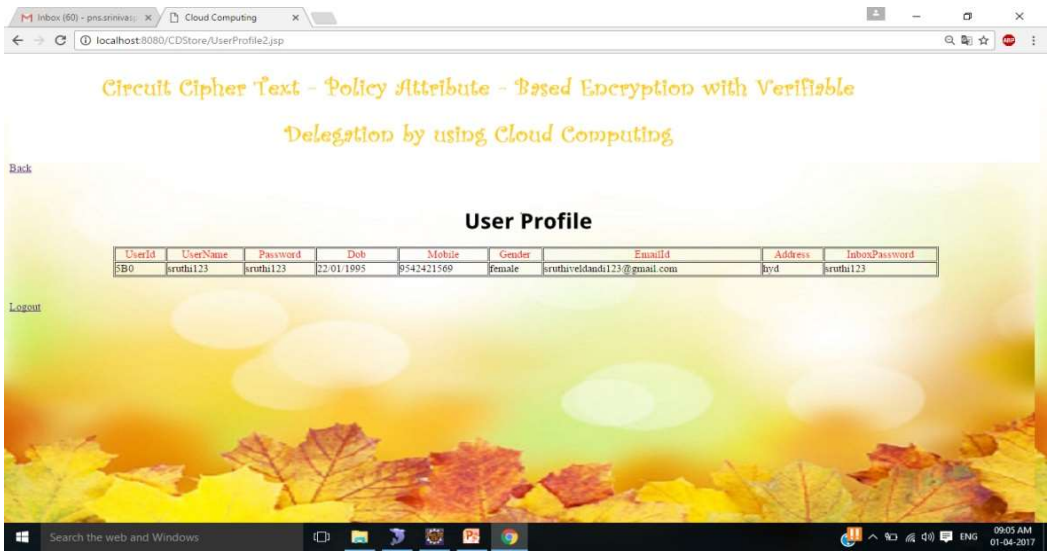
Input: Once the person logs in using the right credentials, our security method will take effect.
Output: Through the usage of our method, users will be able to access cloud services safely and effectively.
Cipher text - policy attribute-based encryption

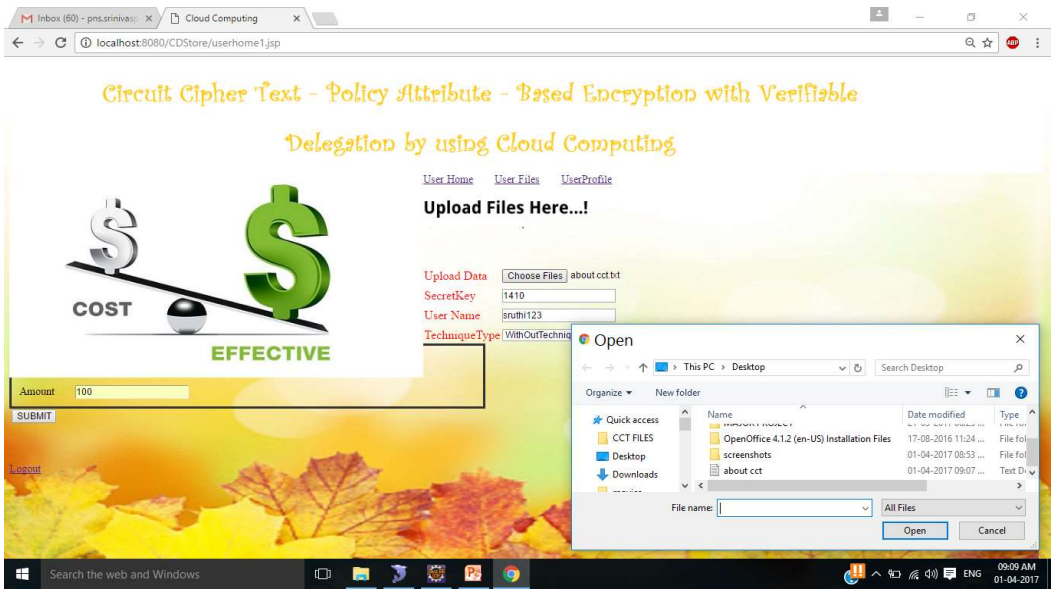
Input: We introduce our hybrid VD-CPABE's definition and security concept. An attribute-based circuit cypher encryption using text policies technique is used in such a system.
Output: We use two-way encryption technique and an encrypt-then-mac approach to provide verifiable delegation, fine-grained access control, and secrecy.
Hybrid Encryption
Input: CD Shop Use hybrid encryption to search the file name.
Output: He will be able to see the user in a hybrid encrypted format. He has to issue a request to the file owner in question in order to access the decrypted file.

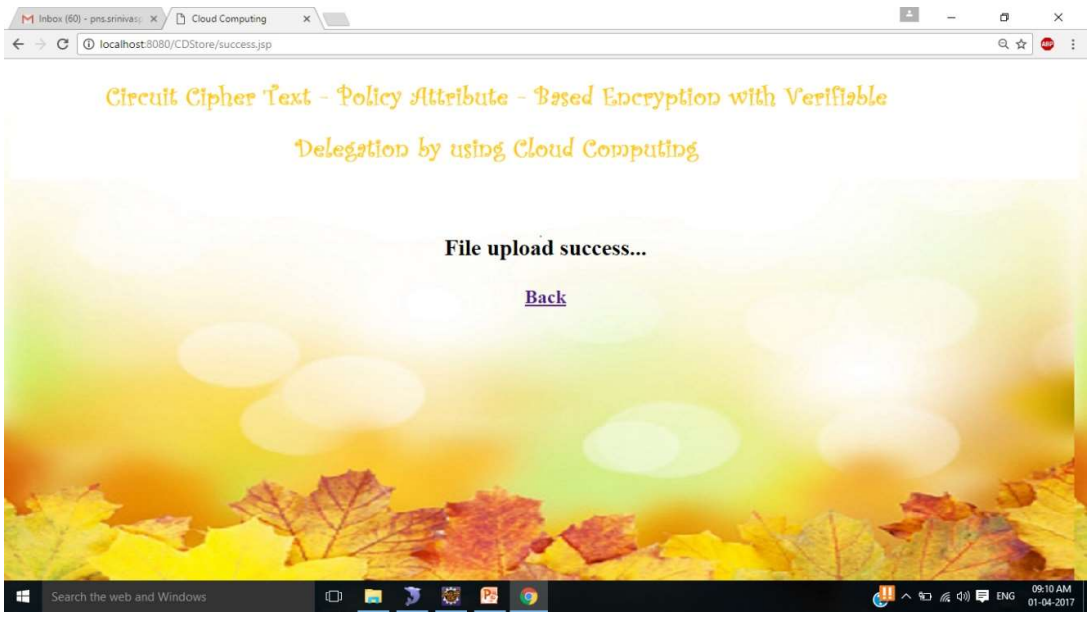
- **Email Authentication**

Input: Send request key to user's email address.
Output: For each file requested, a unique hash code will be produced, and the user will transmit that code to the primary email address of the individual in question.

5.4 SCREENSHOTS:


Cloud Computing
localhost:8080/CDStore/UserProfile2.jsp
Circuit Cipher Text - Policy Attribute - Based Encryption with Verifiable
Delegation by using Cloud Computing
Back
User Profile
UserId | UserName | Password | Dob | Mobile | Gender | EmailId | Address | InboxPassword
5B0 | sruthi123 | sruthi123 | 22/01/1995 | 9542421569 | female | sruthiveldandi123@gmail.com | hyd | sruthi123
Logout
Search the web and Windows
ENG
09:05 AM
01-04-2017


Cloud Computing
localhost:8080/CDStore/userhome1.jsp
Circuit Cipher Text - Policy Attribute - Based Encryption with Verifiable
Delegation by using Cloud Computing
User Home
User Files
UserProfile
Upload Files Here...!
COST
EFFECTIVE
Upload Data
Choose Files
about cct.txt
SecretKey
1410
User Name
sruthi123
Amount
100
SUBMIT
Logout
Open
This PC › Desktop
Search Desktop
Organize
New folder
Quick access
CCT FILES
Desktop
Downloads
Name
Date modified
OpenOffice 4.1.2 (en-US) Installation Files
17-08-2016 11:24 ...
screenshots
01-04-2017 08:53 ...
about cct
01-04-2017 09:07 ...
File name:
All Files
Open
Cancel
Search the web and Windows
ENG
09:09 AM
01-04-2017



SYSTEM TESTING:

Finding mistakes is what testing is all about. The goal of testing is to discover any and all vulnerabilities or faults in a product or service. The method allows for the testing of various parts, assemblies, subassemblies, and final products. Software testing involves making sure the program works as expected and doesn't break anything that users don't want. A variety of tests are available. There is a specific testing requirement that each kind of test aims to meet.

Creating an all-encompassing strategy to assess the unique features and general functioning across different platform combinations is the first step in the testing process. Strict protocols for quality assurance are adhered to.This procedure ensures that the program satisfies all of the requirements laid forth in the system requirements document and is free of bugs. When building the framework, the following factors were taken into account once the testing procedures were established.

CONCLUSION

With a brief mention of the CD Store restoration cost, we focus our analysis on the backup cost in this case. To decrypt the original secrets and restore a backup, a CD Store client downloads slightly more data than what was in the initial backup. The restore cost of CD Store is about the same as the cost of a single cloud system if all clouds have the same external transfer cost. If not, it represents the sum of all k clouds' outgoing transfer costs. Businesses may use CD Store to get into agreements with public cloud providers for archive and backup storage. According to our cost analysis, CD Store saves a significant amount of money by using deduplication.

In further development, we'll consider creating a whole new cloud storage service based on CD Store and addressing other technological issues. Having clients do global deduplication is a natural approach to identify dispersed storage. To be more precise, information fingerprints are made by a client before to data being sent to the cloud.

REFERENCERS

1. M. Green, S. Hohenberger, B. Waters, "Outsourcing the decryption of ABE Ciphertexts", *Proc. USENIX Security Symp.*, pp. 34, 2011.
2. J. Lai, R. H. Deng, C. Guan, J. Weng, "Attribute-based encryption with verifiable outsourced decryption", *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 8, pp. 1343-1354, Aug. 2013.

3. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption Ming Li; Shucheng Yu; Yao Zheng; Kui Ren; Wenjing Lou
4. HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing Zhiguo Wan; Jun'e Liu; Robert H. Deng
5. Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems Junbeom Hur; Dong Kun Noh